REMARKS

Claims 1-33 are currently pending in the application. Claims 1-10, 12-16, 18, and 20-33 were rejected. Claims 11, 17, and 19 were objected to. Claims 5, 10, 13, and 17-20 have been amended. Claims 1-4, 6-9, 12, 15, 16, and 21-33 have been canceled without prejudice.

The Applicant respectfully acknowledges the Examiner's indication of allowable subject matter in claims 11, 17, and 19. As discussed below, amendments have been proposed herein which Applicant believes places these claims in condition for allowance.

The Examiner rejected claims 1, 2, 6-10, 12, 13, 15, 18, 20, 23, 24, 27-30, and 33 under 35 U.S.C. 102(b) as being anticipated by U.S. Patent No. 5,673,388 (Murthi). The Examiner rejected claims 3-5, 14, 16, 21, 22, 25, 26, 31, and 32 under 35 U.S.C. 103(a) as being unpatentable over Murthi in view of a variety of references. Some of the claims have been amended to independent form and some of the rejections are obviated thereby. In addition, the rejections of at least some of the claims are traversed.

Murthi describes a technique for the "[t]esting of shared memory (RAM) in a multiple processor computer system" which "is achieved by partitioning the memory and allocating the memory portions to respective processors in the system. Each processor performs testing of its allocated memory portion simultaneously with the other processors in order to reduce the time required to complete the memory test." See Abstract. Contrary to the Examiner's assertions, Murthi fails to describe or suggest features recited in several of the claims as filed.

For example, claim 5 recites that "the plurality of processors are configured in a plurality of processor clusters" and that "the computer program instructions are operable to cause the boot strap processor to assign *only one* of the processors in each cluster to the corresponding portion of the system memory." That is, only one processor in each multi-processor cluster is assigned the task of testing the memory associated with all of the processors in the cluster. The Examiner correctly states that Murthi does not describe or suggest a multi-cluster system, but points to the

Brock reference as disclosing a computer system having the recited processor clusters and then refers to Murthi at column 3, lines 49-52, as teaching the limitation recited in claim 5. The Applicant respectfully disagrees.

The portion of Murthi to which the Examiner referred describes "assigning to each of a plurality of processors a respective separate portion of a commonly accessible memory," i.e., the general notion of partitioning the memory for testing by more than one processor. It does not, however, describe selecting "only one" processor in multi-processor cluster to test the portion of the system memory corresponding to the cluster (i.e., on behalf of the other processors in the cluster). Neither does Brock teach such a limitation. Claim 5 has therefore been amended to independent form and is believed to be in condition for allowance.

Claim 10 of the present application recites that "the computer program instructions are operable to cause each of the selected processors to update the memory testing progress periodically." In rejecting claim 10, the Examiner referred to column 6, lines 6-12, of Murthi which states that "[t]he BSP determines whether each additional CPU is functional on the basis of whether the BSP was interrupted by the other processor or by its own timer interrupt. This sequence is repeated for each CPU in the system, and information about the number of operating processors in the system is stored in a fixed location in memory."

First, the portion of Murthi to which the Examiner referred describes conventional memory testing using a BSP in which the "testing of the shared memory system 10 is performed exclusively by the bootstrap processor (BSP)." See column 6, lines 20-21. Thus, it is not applicable to the invention recited in claim 10. Further, as described in column 8 of Murthi, "[a]fter completing the memory testing at step 90, the processor then sets the corresponding completion flag." Lines 5-7. "When the BSP finishes testing of its portion of the shared memory it sets the completion flag…and loops until all the completion flags have been set…indicating that every processor has finished memory testing." Lines 16-19.

This passage makes it clear that the way in which Murthi's BSP is notified that the individual processors have completed memory testing is by checking the completion flags which are only set when the corresponding processor has completed testing its assigned portion of memory. By contrast, *each* of the processors recited in claim 10 *periodically* updates its memory testing progress to the bootstrap processor. Because Murthi's processors only set the completion flag when testing is complete they cannot, by definition, be said "to update the memory testing progress periodically" as recited in claim 10. Claim 10 has therefore been amended to independent form and is believed to be in condition for allowance. And because claim 10 is believed allowable, claim 11 is believed to be in condition for allowance without further amendment.

Similarly, claim 13 recites that "the computer program instructions are operable to cause the boot strap processor to periodically update status information corresponding to the progress" "in testing of the system memory by the selected processors." An exemplary embodiment is described in paragraph [0037] of the present application. The Examiner referred to Murthi at column 8, lines 16-19 and again to column 6, lines 6-12, as anticipating this claim.

As stated above, the reference to column 6, lines 6-12, is irrelevant in that it describes the conventional technique in which the BSP performs all of the memory testing. Thus, it is not applicable to a context in which multiple processor are performing memory testing. In addition, while Murthi describes the BSP checking the completion flags in column 8, it does not describe nor suggest the BSP making periodic updates to "status information corresponding to the" memory testing progress made by the other processors. The fact of the matter is that Murthi does not describe or suggest generating or maintaining any such status information. Claim 13 has therefore been amended to independent form and is believed to be in condition for allowance.

Claim 17 has been amended to independent form as suggested by the Examiner and is believed in condition for allowance.

Claim 18 recites that "the computer program instructions are operable to cause the boot strap processor to allocate separate stack memory in a shared memory for each of the selected processors." In rejecting claim 18, the Examiner referred to Murthi at column 7, lines 57-67, which describes how each of Murthi's processors determines the memory range it should test from the sequence number it receives from the BSP. By contrast, the invention recited in claim 18 provides "separate *stack* memory" for *each* of the processors testing portions of the memory. As is well known, stack memory is the memory a processor uses to store and retrieve the code it is currently executing, e.g., the computer program instructions which govern the testing of the memory to which it has been assigned.

As described in paragraph [0039] of the present application, by allocating separate stack memory for each processor, it is ensured "that the memory testing threads being simultaneously executed by the multiple processors do not interfere with each other." This is to be distinguished from simply determining the memory range being tested by the processor which is what Murthi is describing in the portion of column 7 to which the Examiner referred. Presumably this determination is controlled by instructions in a stack memory somewhere in the system, but Murthi does not mention such a stack memory and clearly does not describe or suggest the allocation of separate stack memory for each processor. Claim 18 has therefore been amended to independent form and is believed to be in condition for allowance.

Claim 19 has been amended to independent form as suggested by the Examiner and is believed in condition for allowance.

Claim 20 recites that "the computer program instructions are operable to associate a lock prefix with instructions targeting a shared memory associated with the boot strap processor thereby ensuring that two of the processors do not access the shared memory at the same time." In rejecting claim 20, the Examiner referred to Murthi at column 3, lines 10-20. The portion of Murthi to which the Examiner referred describes a "time division multiplexed communication

between the plurality of processors and the memory controller." As described, this allows each of the processors to communicate with its portion of the memory through the memory controller during its assigned time slot.

By contrast, the "lock prefix" recited in claim 20 should not be equated with the notion of time slots inherent in a time division multiplexing (TDM) scheme such as that to which Murthi refers. Rather, this alternative mechanism simply operates to ensure "that two of the processors do not access the shared memory at the same time" by associating the prefix with "with instructions targeting a shared memory." While such a mechanism could conceivably be used to implement a TDM, it is not anticipated by the existence of TDM. That is, other mechanisms such as, for example, providing system bus access to each processor during its corresponding time slot, are more likely to be used. In addition, the approach recited in claim 20 may also be used to simply allow access to memory in the order in which memory requests are received, further distinguishing the claimed invention from TDM.

Moreover, claim 20 recites that the shared memory (as distinguished from the "system memory" recited in the claims) is "associated with the bootstrap processor." This is clearly not the case with Murthi in that the memory being accessed by the various processors is the memory being tested which is not specifically "associated with the bootstrap processor." Claim 20 has therefore been amended to independent form and is believed to be in condition for allowance.

CONCLUSION

In view of the foregoing, Applicants believe all claims now pending in this application are in condition for allowance. The issuance of a formal Notice of Allowance at an early date is respectfully requested. If the Examiner believes a telephone conference would expedite prosecution of this application, please telephone the undersigned at (510) 663-1100.

Respectfully submitted,
BEYER WEAVER & THOMAS, LLP

/Joseph M Villeneuve/
Joseph M. Villeneuve
Reg. No. 37,460

P.O. Box 70250
Oakland, California 94612-0250
(510) 663-1100